

## AMENDMENTS TO THE SPECIFICATION

On page 7 of the specification, please replace the paragraph stating “In the preferred embodiments of the present invention . . .” with the following paragraph:

“In preferred embodiment of the present invention, the run-time space efficiency of standard JavaCard technology is combined with the flexibility of the standard Java linking scheme. Referring to Figure 2, in preferred embodiments of the present invention, standard Java class files of a Java program are pre-processed and converted by a cap file generator to JavaCard cap files 10, wherein the constant pools 30 contain the original Java symbolic linking strings for externally and internally referenced items. Further, an adapted JavaCard VM 55 is used including an additional converter 100. During the downloading and linking process, the converter 100 of the adapted JavaCard VM 55 executes a mathematical algorithm, so-called hash function, on the standard symbolic linking strings 80 of the ~~constant pool 20~~ constant pool 30 of the downloaded cap file 10. The hash function 100 generates, for each symbolic linking string 80, a short token 65, preferably a number. These tokens 65 are used to replace the symbolic linking strings 80 in the constant pool 20 of the cap file 10 and are further stored in the export table 40 of the Adapted JavaCard VM 55. During the linking process the adapted JavaCard VM 55 looks up the referenced items in the bytecode instructions by the tokens 55 generated with the hash function and then replaces these references by the corresponding run-time specific identifiers.”

On page 8 of the specification, please replace the paragraph stating “In the preferred embodiments of the present invention . . .” with the following paragraph:

“In preferred embodiments of the present invention, the converter 100 of the adapted JavaCard VM 55 employs a parameterized hash function to map the symbolic linking strings 80 of the ~~constant pool 20~~ constant pool 30 of the cap file 10 to associated tokens 65. This parameter is preferably calculated by the cap file generator, i.e. the Java compiler, which translates the Java source file in the corresponding cap file format 10 suitable for downloading and executing in embedded microcontrollers. The parameter which is stored in the ~~constant pool 20~~ constant pool 30 of the cap file 10 to be downloaded on the adapted JavaCard VM 55 is used by the parameterized hash function to avoid generation of the same token 65 for two different symbolic linking strings 80. The parameter supplied with the ~~constant pool 20~~ constant pool 30 guarantees that two symbolic linking strings 80 in the ~~constant pool 20~~ constant pool 30 naming two different items will never be mapped to the same token 65 by the parameterized hash function.”

On page 8 of the specification, please replace the paragraph stating “The parameter for the constant pool 20 . . .” with the following paragraph:

“The parameter for the ~~constant pool 20~~ constant pool 30 of the cap file 10 may be calculated by the cap file generator as follows. The cap file generator may check the symbolic linking strings 80 included in the ~~constant pool 20~~ constant pool 30 of the cap file 10 and varies a start parameter until it finds a parameter which satisfies the condition that the hash function to be used in the adapted JavaCard VM 55 maps all symbolic linking strings 80 on different tokens 65. To create a parameter for a JavaCard cap file 10, the cap file generator only needs to know the

corresponding hash function used in the JavaCard VM 55.”

On page 9 of the specification, please replace the paragraph stating “Dependent on the number of different constants . . .” with the following paragraph:

“Dependent on the number of different constants used, there may be strings 80 which are mapped to the same numbers. If such a string 80 occurs, at least one of the constants of the hash function may be varied by the cap file generator to avoid a number clash. The varied constant may then be stored as a parameter together with the strings 80 in the ~~constant pool 20~~ constant pool 30 of the cap file 10 to be used by the converter 100 of the JavaCard VM to correctly map the strings 80 in the ~~constant pool 20~~ constant pool 30 of the cap file 10 to numbers.”

#### **AMENDMENTS TO THE DRAWINGS**

Please replace FIG. 1 with the replacement sheet for FIG. 1 filed herewith.